A Design and Control Environment for Internet-Based Telerobotics

Roberto Oboe* and **Paolo Fiorini†**

* Dipartimento di Elettronica e Informatica
Universitá di Padova
Padova, 35131 Italy

† Jet Propulsion Laboratory
California Institute of Technology
Pasadena, CA 91109 USA

Abstract

This paper describes an environment for the design, simulation and control of Internet-based force-reflecting telerobotic systems. We define these systems as using a segment of the computer network to connect the master to the slave. Computer networks introduce a time-delay that is best described by a time-varying random process. Thus, known techniques for controlling time-delay telerobots are not directly applicable, and an environment for iterative design-and-test is necessary. The underlying software architecture supports tools for modeling the delay of the computer network, designing a stable controller, simulating the performance of a telerobotic system, and testing the control algorithms using a force-reflecting input device. Furthermore, this set-up provides data about including Internet into more general telerobotic control architectures. To demonstrate the features of this environment, the complete procedure for the design of a telerobotic controller is discussed. First, the delay parameters of an Internet segment are identified by probing the network. Then, these parameters are used in the design of a controller which includes a quasi-optimal estimator to compensate small data losses. Finally, simulations of the complete telerobotic system and emulations using a planar force-reflecting master and a virtual slave exemplify a typical design and test sequence.

1. Introduction

This paper describes a computer environment for the design, simulation and test of control algorithms for Internet-based telerobotic systems. We define these systems as those using a segment of Internet to exchange data between the master and the slave of the telerobot. In particular, we are interested in the class of *force-reflecting* telerobots with which the operator can feel the forces applied by the remote robot to its environment.

This class of systems is potentially very important, since the avail ability of Internet-based telerobots would stimulate the design of new int eractive applications on Internet-capable computers. We refer to the possibility of' enabling Internet users to exchange proprioceptive, tactile and kinesthetic data over the computer network. These data, loosely defined as *haptic* information, would provide a realistic feeling of' telepresence and the ability of touching remote objects and people. Applications of this technology would include health care, entertainment, remote training, and collaborative work over a shared environment.

Only a few years ago, data transfers among computers were limited to text-1)asd formats such as electronic mail and *talk* sessions. Now, continuous improvements of Internet performance and powerful new applications have made the exchange of multimedia data very easy. Internet users can receive audio and video data with near real-time performance, as demonstrated by recent teleconfer encing software packages [3,26]. However, the use of haptic devices over the Internet is a much more challenging problem than transferring audio and video data, since these devices must remain stable in spite of the performance fluctuations typical of Internet. In particular, Internet data have random time-dday and packet losses which depend on the characteristics of the network and on its load. Current control design tools are inadequate for Internet-based force-reflecting telerobots, since they lack identification and analysis procedures to address the specific characteristics of Internet time-delay.

Past work on environments for telerobotics control has focused primarily on modeling [19,12], real-tiltlc architectures [31,17], and control [1] issues. In [19], a position-based force-reflectin g system is modeled using electrical components in a *Spice* simulat ion environment. A more flexible modeling approach is presented in [12] using a library of modules for the simulation program *Matlab*. In [31], an example is presented of a real-time hardware and software architecture supporting telerobotic control and consisting of a *VME* chassis running the *Chimera 11* operating system. A software architecture for controlling a dexterous manipulator using a personal computer is described in [17], showing the use of consumer technology in robotic control. A powerful modeling and control architecture for telerol botic control is presented in [1]. This environm ent consists of a series of software modules, based on commercial hardware and software, that can be combined to model and control an ample array of robotic devices. This approach achieves global stability for the telerobot by ensuring that each component meets a criterion based on passivity theory. The control of tilllc-delay telerobotic devices has mostly addressed the case of constant delay. The main approaches used include passivity theory [2], remote compliance control [21], and wave variables decomposit ion [27]. Some of t hese methods achieve *independent of delay* stability (IOD), as shown in [15]. However, they arc not directly applicable to the variable time-delay of Internet, since a controller designed for a fixed delay $\tau = T$, may not stabilize the system when the delay is a variabl e $\tau(t), 0 \leq \tau(t) \leq T$, as demonstrated in [20].

So far, of the two main prol blems affecting Internet communication, i.e. variable time-delay and data packet 1osscs, only the former has received sufficient attention. Variable ti~nc-delay is addressed mostly by estimating the original data [23], or by including the delay variations directly into the design procedure [1 O,25,32,33]. In [23], the variation of the delay is compensated using an *n-step* predictor, which also recovers from data

packet losses. This approach is quite conservative, since it generates a controller tuned to the average communication delay. The variable time-delay can be used directly in the design as an uncertain parameter and compensated using robust control design techniques. Systems verifying the *matching conditions* of uncertainty [22] are discussed in [10,25,32], where it is shown that stable controllers can be designed when the upper bound of the first derivative of the delay is known. Unfortunately, the instantaneous and the delayed state models of a telerobot cannot be decomposed according to the matching conditions. An approach for systems with unmatched uncertainties is presented in [10,33]. However, also in this case there is no solution for a telerobotic system. Adaptive control is proposed for variable time-delay systems in [30], but the fast variation of the delay prevents the application of this technique to Internet control.

For the second problem of Internet communication, i.e. data losses due to packet discard, no remedy currently exists. The normal approach is to require the retransmission of the lost packets, but this is not applicable to the case of Internet-based control. This issue has been addressed in [24] by showing that some packet loss can be compensated by using an *n-step* predictor. However, this approach requires the knowledge of the models of the local and remote processes, and its robustness has not been demonstrated.

From this brief summary, it follows then that one key issue of variable time-delay compensation is the knowledge of some properties of the delay variation), i.e. the availability of a model of the Internet segment between the two parts of the telerobot. Internet is a strongly connected network of computers, communicating with each other using packet-switched protocols [11]. In [5] it is shown that the delay affecting the data packets depends on the packet's routes, on the different handling policies at each node traversed, and on the network congestion. Thus, it is almost impossible to determine an exact, analytical model of an Internet communication. An accepted approximate model consists of a network of queues, one for each node along the path connecting two computers [18]. According to this model, queuing and dynamic routing introduce the variable transmission delay, and network congestion may result in packets losses. Thus, the approximate Internet model is characterized by the statistics of the packet delay and of the packet losses. Values of these parameters have been known for some time [4] but, because of the rapid Internet growth and variable loading conditions, they need frequent updates and on-line identification [16,28].

In this paper, we propose a simple architecture that implements a solution to the three main issues of Internet-based control, namely the identification of the delay properties of an Internet segment, the design and analysis of a telerobotic controller, and the test of the controller with a planar force-reflecting master interacting with virtual objects. We show the validity of this approach by presenting the results of each phase, and discuss the indications that this architecture provides for more general telerobotic systems.

The paper is organized as follows. Section 2 gives an overall description of the system architecture. The end-to-end performance analysis of an Internet segment in different operating conditions is presented in Section 3. Section 4 presents a new controller design for an Internet-based telerobotic system. Section 5 describes some simulations and experiments carried out with this system, Finally, Section 6 concludes the paper and presents our plans for future work in this area.

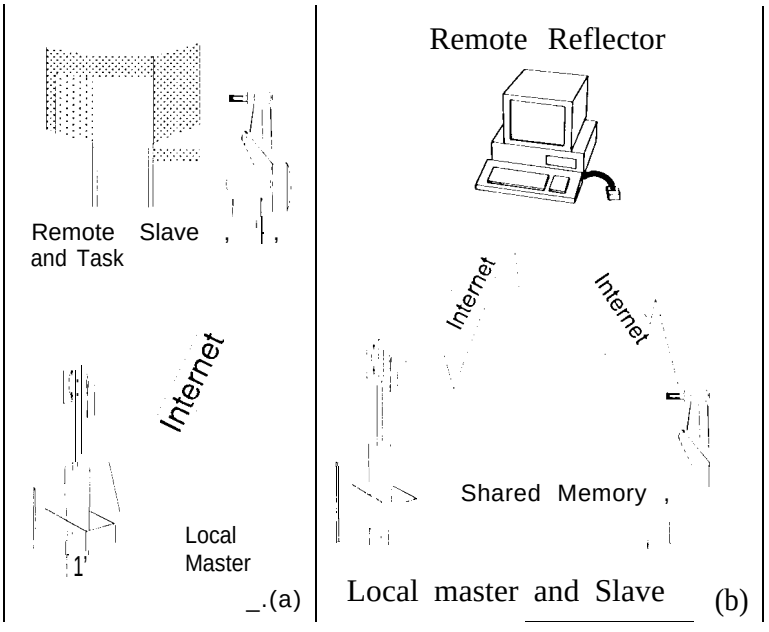

Figure 1: (a) A typical structure of' Internet-based telerobot its. (b) The simplified structure used in this environment.

2. System architecture

In this Section we describe the architecture of this implementation of the design and control environment for Internet-based telerobotic systems. In general, a telerobotic system consists of two components, the master and the slave, connected by a communication line. Here, we address the case of the communication between master and slave being carried by a segment of Internet, as shown schematically in Figure 1-(a). The Figure shows a telerobotic system [13] consisting of a six degree-of-freedom (dof) force-reflecting master and a six-dof slave manipulator connected by Internet. To keep the logistics of' the system manageable, we use a *reflector*, i.e. a remote computer whose function is to echo the received data to a different computer, as shown schematically in Figure 1-(b). With this approach, the master can be located next to the the slave of the telerobot, and the communication line connecting the two can be a segment of Internet with suitable characteristics. For example, in the tests described later, we examine the effects of different time-delays by selecting reflectors located at different distances.

Although simple from an implementation point of view, this approach allows a significant degree of flexibility in configuring the experimental set-up. We take advantage of this by using one personal computer (I 'C) for the design of the controller, and a second PC to control the telerobot. One computer supports the identification and design components of the environment, and the other PC is the real-time controller, as shown schematically in Figure 2. The identification and design uses the *Matlab-Simulink* program within a popular desktop environment, and is interfaced to a set of dedicated routines for Internet modeling. The mal-time controller takes advantage of the accessibility of DOS interrupts.

The interface among the various parts of the environment consists of shared data

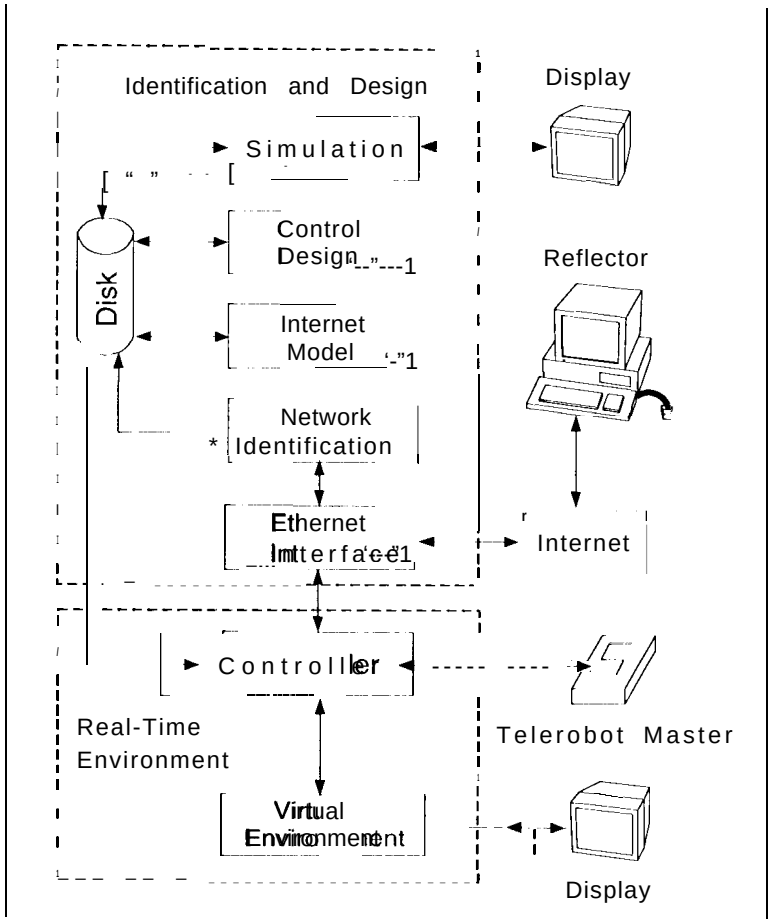

Figure 2: The design and control environment for Internet-based telerobotics.

files. The environment can also be implemented in a single-computer configuration, in which design and control programs reside on the same machine and operate sequentially. In the dual-computer configuration described here, the two functions are carried out in parallel. This configuration can be generalized to more complex telerobotic systems, by using true PC-based robot controllers [17] as a replacement of the virtual slave.

In the following Sections, the two parts of the environment of the dual computer configuration are described in some detail.

2.1. The identification and design

The identification procedure is shown in the top part of Figure 2. During this phase of the controller design, the PC sends probing packets to the slave telerobot via the reflector, and measures the round trip time. Packets length and rate are chosen by the user, according to the target application. The parameters of the time-delay between the two computers and the statistics of the packet losses are stored in a file and a *Matlab* procedure computes the approximate model of the connection delay and displays the results, as discussed later in Section 3. If the Internet connection satisfies the requirement

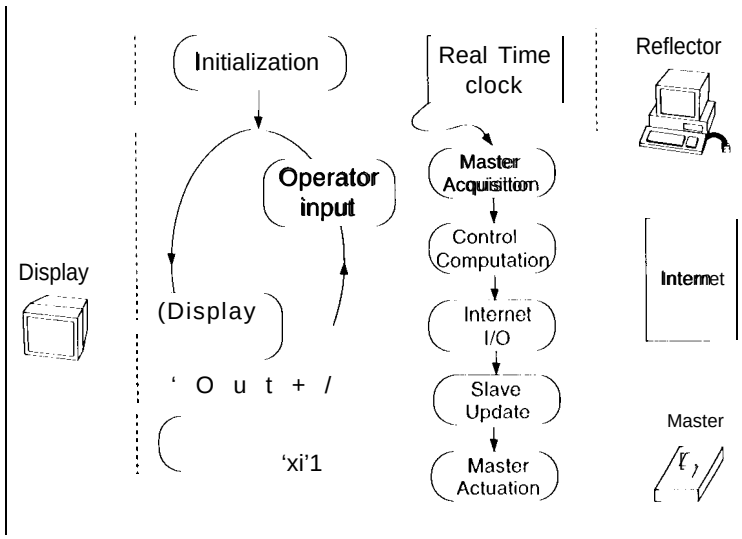

Figure 3: The real-time environment.

of' time-delay and packet losses, the user can proceed with the controller design and analysis.

The implications of designing and simulating a telerobotic controller connected to Internet arc minimized by considering the Internet as an external *black box* connected to the computer. Thus the procedure described in Section 3 is equivalent to the identification of a hardware device interfaced to the design environment.

The design procedure is also shown in the top part of Figure 2, and consists of a *Matlab* module computing state feedback and observer gains, as described later in Section 4, and of a *Simulink* model Of the completesystem, simulating the telerobot performance using the true delay data. When the results of the simulation are satisfactory, the controller is ported to the real-tilllc environment described next.

2.2. **The real-time environment**

The test of the control algorithms is shown in the bottom par t of Figure 2, and consists of the execution of simple telemanipulation experiments. 111 order to save development time, the telerobot used here is the *Pantomouse* described in [8]. This system consists of a controller for a two-dof, force reflecting planar manipulator, and of a virtual environment populated by several objects. These two elements represent the local master and slave shown in Figure 1-(1)). One of the objects is defined as the end effector of a virtual slave manipulator, controlled in position and force by the planar manipulator which then becomes the telerobot master. '1'11(! 1 'ant omouse emulates a complete telerobot by computing in real-tiltlc the interaction of the slave manipulator with the objects in the virtual environment, and by applying the resulting virtual forces to the actuators of the two-dof master. The physical characteristics of the virt ual objects (position, shape, mass, friction, etc.) arc user defined, and the interaction between the virtual slave robot and the virtual objects is displayed on the computer screen. By using a virtual slave, both master

Host	Distance (Km)	Average Delay (ms)	Standard deviation	Loss rate
1. Local	0.05	0.998	0.715	0.00
2. Same Domain	30	8.10	5.35	0.08
3. Different City	150	17.20	9.74	0.80
4. Different Continent	10000	326.3	27.20	41.4

Table 1: t,iln(!-delay parameters for typical Internet connections

and slave controller arc implemented in the same computer, thus greatly simplifying the experiments. I lowever, the need of computing in real-time the evolution of the virtual environment limits the number of allowable objects interacting with t he slave robot.

The controller for the telerobot system can be implemented with a centralized or a decentralized structure. The first approach is well suitable for this architecture, but it dots not address the reliability requirements of real systems, since in case of communication interruptions, either the master or the slave will remain without control. A decentralized controller is more robust to communication interruptions and is described later in Section 4.

The real-tlilnc software is described in [8] and is summarized here for completeness. It has a *two-level* structure', in which a main program handles all housekeeping functions, and a rml-time interrupt server performs the time critical activities, as shown schematically in Figure 3. I lousekeeping activities are carried out at a lower priority than the control, and include initialization of the virtual environment, user interface and graphical display. Time critical activities include data acquisition from the master, computation of the slave commands, communication with the reflector, comput ation of the int erac tion forces in the virtual world, and actuation of the master.

Internet across is implemented with standard socket datagrams that arc sent to, and received from, the reflector within the real-time program. This approach allows different types of time-delay, since the communication with the reflector can be replaced by other delays such as, for example, a fixed delay generated by a *first-in-first- out* memory buffer. In this system, the forward data path is routed through the remote *reflect or*, whereas feedback data, from the slave to t he master, are exchanged within t he real-time soft ware. This data flow structure is justified by the assumptions summarized in Section 4.

In the following Sections, we describe the various functions of the system and justify our architectural choices. Furthermore, we discuss simulations and experiments of a position-based force-reflecting cont roller designed using this environment.

3. Internet modeling

In this Section, we discuss the charact eristics of Int ernet that arc relevant to a telerobotic controller. As mentioned in Section 1, the two main factors influencing Internet-based robotic control are the transmission delay between the master and the slave, and the 10SS Of data packets due to network congestion.

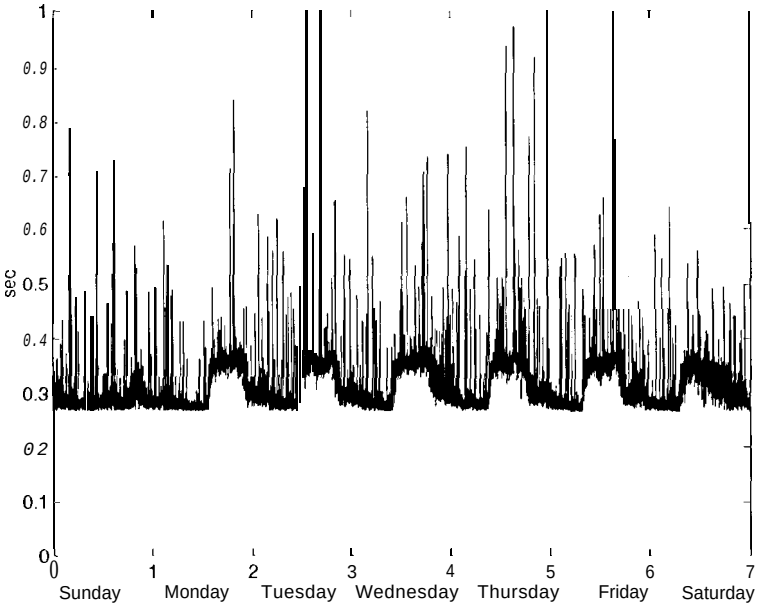

Figure 4: Weekly variation of the delay

We treat the Internet segment as a black box, since it is very difficult to generate an analytical model of the (I('lay, and we use software tools to measure, over a given time interval, the round trip time (RTT) of probing packets sent to the remote telerobot. The modeling difficulty stems from the large number of Internet users and from the throughput, queues and routing policy of each computer along the connection. In fact, before reaching their destination data packets traverse several computers, called *nodes*, each handling data streams from different sources. Each node queues the incoming packets and routes them to a node closer to the packet destination. If a node is overloaded, it may discard some of the incoming packets, or it may route them to a less loaded server, thus introducing an unpredictable delay in the data stream, and possibly loosing some packet [16].

The standard tools of Internet communication are based on TCP, UDP and ICMP protocols [1 1]. The latter is used only for network monitoring purposes, TCP includes reliability features, such as error recovery, large packet splitting and reordering, and UDP establishes a bare data connection between two processes. Since TCP features introduce an excessive overhead and cannot be excluded by the user, the UDP protocol is preferred for constant data-flow applications such as control and multimedia sessions. We use an ICMP-based tool to measure 1{ '1 ''1', since it adds a negligible overhead to the bare packet trawling time, and therefore yields a true measure of the network performance.

The *reflector* simplifies the computation of the 1{"1''1', since all measurement are done locally in the computer transmitting the probing packets. The ICMP-based tool is a modified *ping* routine, which sends a stream of probing packets to the reflector [7]. Packet rates from 10 to 1 ()() *ms* are chosen, since they are close to typical telerobotic communication rates. Table 1 shows the delay parameters of connections of various lengths measured with 100 *ms* probes. The duration of the measurement is approximately 1000 s and it

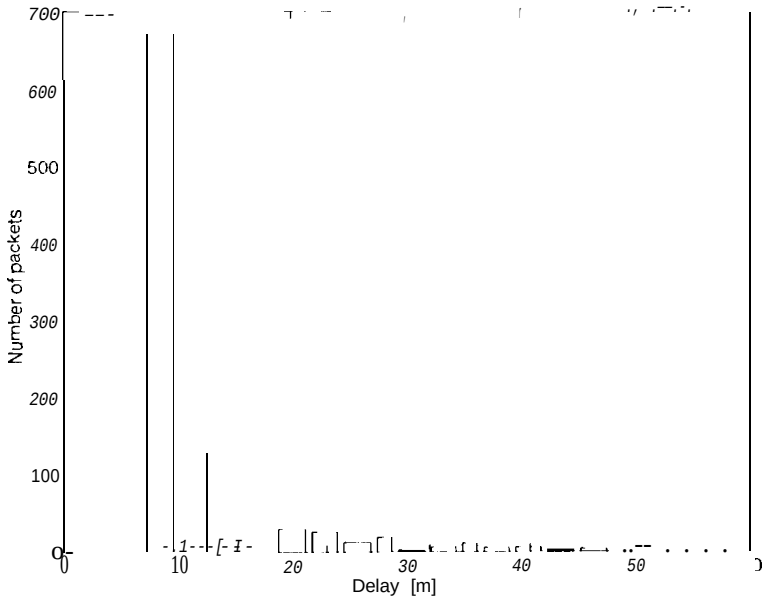

Figure 5: 1)elay for 150 km connection.

has been performed during regular office hours. Table 1 shows no clear relation between the average delay of a packet and the distance between the source and the destination computers. Clearly the time-delay increases with distance, but t 11(! delay depends also on the number of nodes traversed, on the speed of each segment and on th e communication policies of each node.

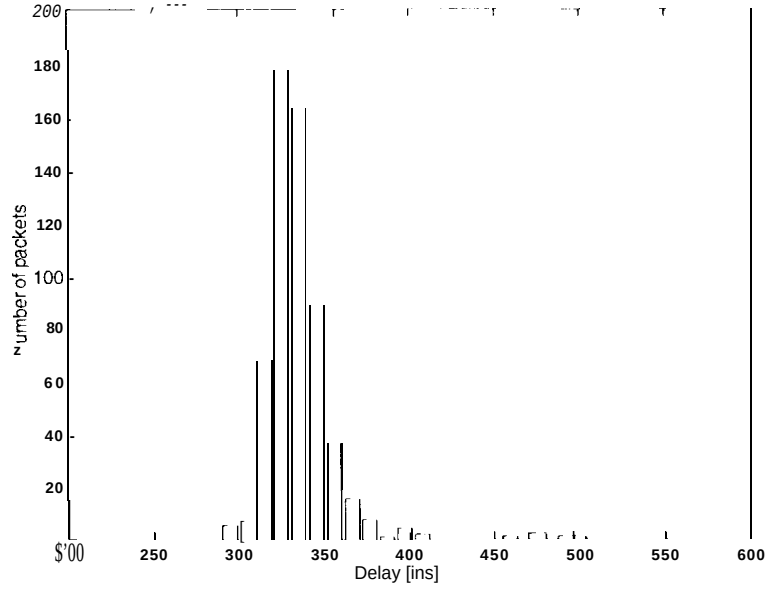

Figure 6: 1)elay for 1 0000 km connection.

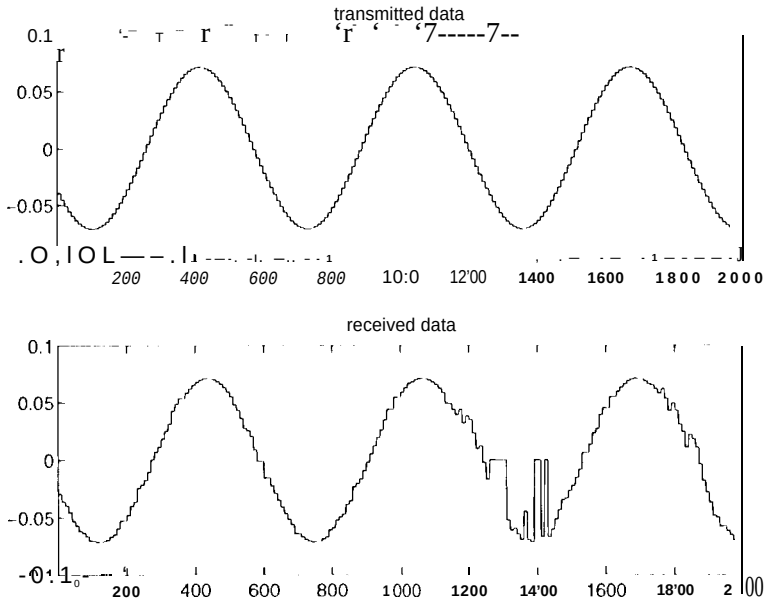

Figure 7: Effects of variable delay and losses on signals.

The average delay strongly depends on the network load and usually shows daily and weekly variations, as shown in Figure 4. In the future, the daily fluctuation of the delay can be used to predict the delay parameters and adapt the controller gains. A more detailed description of the measurements performed is found in [28].

From these measurements, one can compute the throughput of" the connection [4] and define the upper limit of the controller bandwidth. Since it is shown in [4] that by limiting the data rate to 10% of the throughput the network is not overloaded, the design of' an Internet-based telerobot must find a compromise between the data rate and the resulting packet delay. In principle, a higher rate would improve the performance of" the complete system, however the resulting network overload" would slightly increase the delay and substantially increase the packets losses on the network segment, thus reducing the actual system performance.

The shape of the probability distribution of the time-delay and its variance depend on the number of nodes traversed. If the connections spans only a few nodes, the distribution is exponential, since the dominant factor is the performance of the slowest node, as shown for Host 2 in Figure 5. However, if the number of nodes is sufficiently large, the time-delay follows the Gaussian distribution of Figure 6, as it is expected in a network of independent queues such as, for example, a connection between the University of Padova and the Jet Propulsion Laboratory [] 8]. Long distance connections imply larger packet delays and larger variations of the delay, since data packets must access crowded 1)ack-1)011(: segments. Furthermore, Table 1 shows that higher packet losses are to be expected even for low data rates when the connection involves an intercontinental segment.

The effects of the variable time-delay and of the packet losses are shown in Figure 7 for a test waveform sent over Internet. The upper part of the Figure shows a sinusoidal signal sampled at 10 *ms* rate, and the bottom part of" the Figure shows the waveform

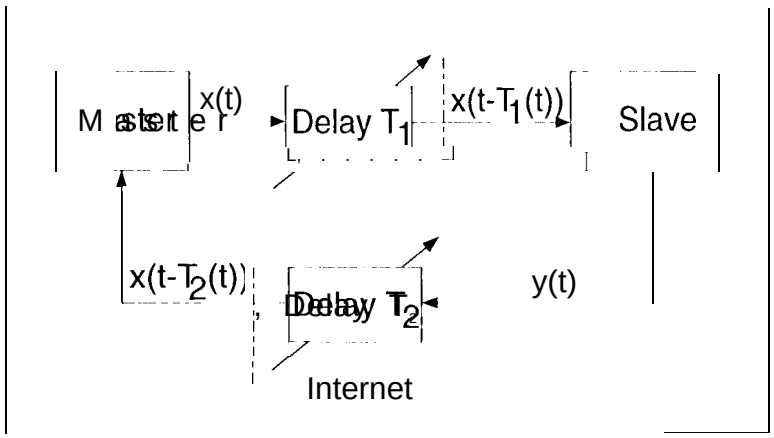

Figure 8: Internet-based telerobot.

received at the remote side of a 1 50 Km Internet segment. The time delay variations and packet losses appear as noise on the original sinusoid, and are perceived by the user as a feeling of roughness in the force feedback. The spikes in the waveform represent lost packets, whereas the waveform distortion is due to the variable time-delay, which alters the packets sequence.

Because of these considerations, in the near future Internet-based telerobotic control architectures should be designed mostly for small geographical areas, served by local area networks with a small number of nodes and users. This limitation does not preclude a number of interesting and useful applications, such as, for example, tele-surgery within the same hospital, and home care within a single city. In the longer term, the availability of higher Internet bandwidth will extend telerobotic control to larger areas. Furthermore, additional improvements are expected from two new Internet standards, ReSerVation Protocol (RSVP) and Internet Protocol version 6 (IPv6). RSVP is a resource setup protocol that supplements the basic IP service [6] by allocating specified values of throughput and maximum delay. IPv6 extends Internet addressing space and thus, by simplifying routing, it should reduce the data delivery time [14]. However, the time-delay will still be variable, and Internet-based control will still require specific techniques.

After this brief summary of the main problems due to Internet within a telerobotic system, in the next Section we describe a procedure for designing a controller that compensates the time-delay variation and the packet loss.

4. Controller design for an Internet-based telerobot

To make the control problem more manageable, we make a few simplifying assumptions. In general, the forward and the feedback data paths of an Internet-based telerobotic system are characterized by different delays, $T_1(t)$ and $T_2(t)$ with $RTT(t) = T_1(t) + T_2(t)$, and by different packet flows, as shown schematically in Figure 8. However, if we assume that master and slave are linearized by suitable controllers, we can simplify the model by collapsing the two delays into a single one, arbitrarily located in the forward

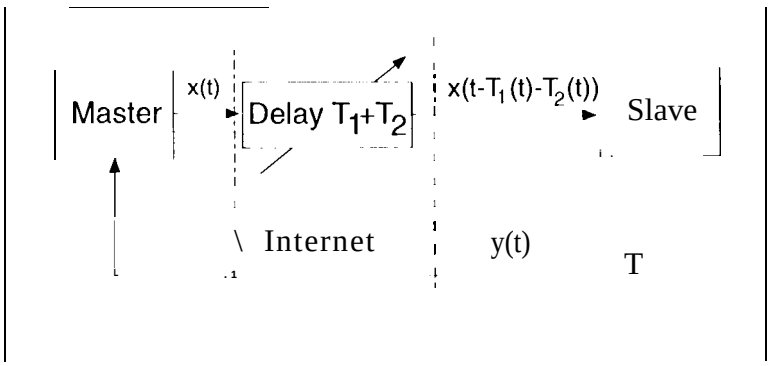

Figure 9: Simplified model of Internet-based telerobot.

direction. Furthermore, by assuming that both data paths are routed through the same Internet segment, we consider RTT consisting of two identical components $h(t)$, with $h(t) = \mathrm{RTT}(t)/2$. Similarly, we assume that packet losses are equally distributed on each segment. Given these assumptions, the model of Figure 8 becomes the simplified model shown in Figure 9. In general, this model is only an approximation of a telerobotic system, but here it, justifies the structure of the data paths used in the real-time controller described in Section 2.2. Data from the master to the slave are send through the Internet via the remote *reflector*, and data from the slave to the master are exchanged internally to the PC in the real-tlilnc controller. Therefore, the total communication delay Ii'l"l' affects only one data path, as in the approximate telerobot model shown in Figure 9. Furthermore, the implementation of the slave within a virtual environment satisfies the linearity assumption for the slave, and the mechanical structure Of the master guarantees a good linearity.

To better represent the architecture used for the real-time environment, the model of Figure 9 must also include the influence of the discrete data communication. Because of the structure of the real-time controller, packets sent by the master to the slave are initially synchronized to the controller cycle time. However, due the variable time-delay introduced by the Internet segment and by the lack of synchronization between the real-time computer and the reflector, the received packets arrive randomly with respect to the control cycle. Since data packets are accepted only at a specific point of the control loop, the variable time-(lday is rounded to a multiple of the controller cycle. This is modeled by adding the two synchronized *sample and hold* elements shown in Figure lo, with a sampling time equal to the controller cycle time.

The results of the Internet measurements show that the time-delay of a connection consists of a slow variation due to the average network load with a period of twenty four hours, and a faster, jitter type, variation due to the nodes of the segment and to the synchronization with the control cycle. In the following Sections we present the design of a controller stabilizing our telerobotic set-up against these variation of the tilll(!-delay. The rapid variations of the time-delay and the packet losses are treated separately as noise affecting the signals, and compensated using a quasi-optimal estimator.

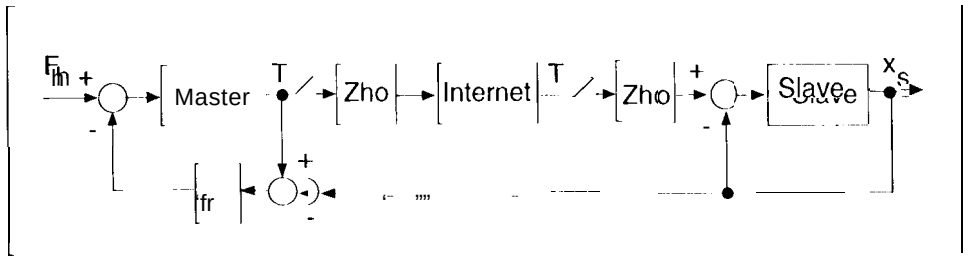

Figure 1 0: Sampled data telerobotic system.

4.1. Controller design

We consider the standard position-based force feedback scheme [1 3] shown in Figure 11, and we in'olmsc a new decentralized controller based on state variables feedback. In this scheme, the forces acting on the master are proportional to the difference between the position of the master and that of the slave. For simplicity, we consider a single dof system, whose sl)acc-state equations arc given by:

$$\begin{aligned} \Sigma_1 \quad &: \quad \dot{x}_1 = A_1 x_1(t) + B_1 u_1(t) + A_{21} x_2(t - h(t)) \\ \Sigma_2 \quad &: \quad \dot{x}_2 = A_2 x_2(t) + B_2 u_2(t) + A_{12} x_1(t - h(t)) \end{aligned} \tag{1}$$

where $h(t) = \frac{T_1(t) + T_2(t)}{2}$, Σ_1 and Σ_2 represent the master and the slave, x_1 and x_2 represent the full state of master and slave, $x_1 = [x_m, \dot{x}_m]^T$ and $x_2 = [x_s, \dot{x}_s]^T$, and u_1, u_2 are the inputs to the mast er and to the slave, respectively. The matrix coefficients of the state equations are given by:

$$A_1 = \begin{bmatrix} 0 & 1 \\ 0 & -\frac{B_m}{M_m} \end{bmatrix} ; \quad B_1 = \begin{bmatrix} 0 \\ \frac{1}{M_m} \end{bmatrix} ; \quad A_{21} = \begin{bmatrix} 0 & 0 \\ \frac{K_{fr}}{M_m} & 0 \end{bmatrix} \tag{2}$$

$$A_2 = \begin{bmatrix} 0 & 1 \\ 0 & -\frac{B_s}{M_s} \end{bmatrix} ; \quad B_2 = \begin{bmatrix} 0 \\ \frac{1}{M_s} \end{bmatrix} ; \quad A_{12} = \begin{bmatrix} 0 & 0 \\ -\frac{K_p}{M_s} & 0 \end{bmatrix} \tag{3}$$

where K_{fr} represents the force feedback gain 1, K_p represents the gain of the slave controller, and M_m, B_m, M_s, B_s represent masses and friction coefficients of the mast er and the slave.

For the system represented by equations (2) we propose the decentralized state feedback controller given by the following equat ions:

$$\begin{cases} u_1 = K_1 x_1 \\ u_2 = K_2 x_2 \end{cases} \tag{4}$$

where $K_1 = [K_1(1), K_1(2)]$ and $K_2 = [K_2(1), K_2(2)]$ are two gain vectors, as shown in Figure 12.

Since state feedback ensures correct t racking only when reference and feedback are multiplied by the same gain, it follows that A_{21} and A_{12} depend on the controller gains:

$$A_{21} = \begin{bmatrix} 0 & 0 \\ \frac{K_1(1)}{M_m} & 0 \end{bmatrix} ; \quad A_{21} = \begin{bmatrix} 0 & 0 \\ \frac{K_2(1)}{M_s} & 0 \end{bmatrix} \tag{5}$$

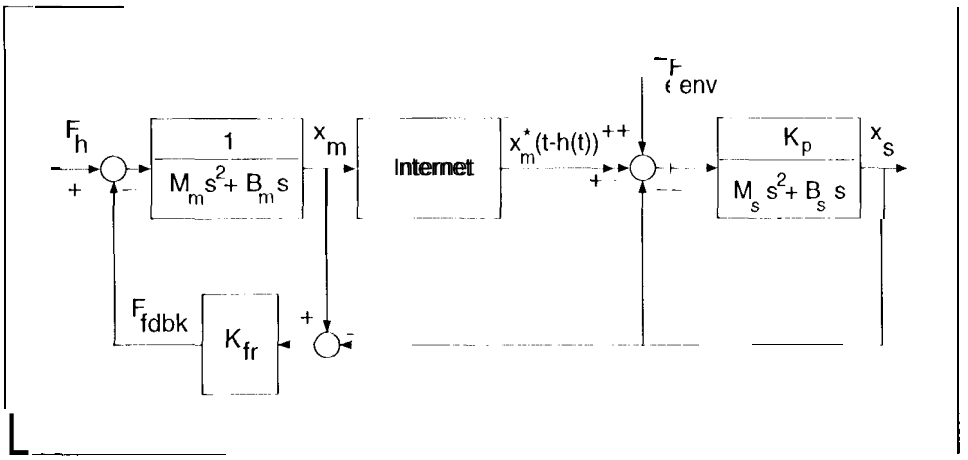

Figure 11: 1 'osition-based force feedback.

Note that this constraint makes solution methods such as the one descr bed in [33] inapplicable to a position-based force reflecting telerobot.

The state equations (2) are rewritten in compact form including the controller equations (4) as:

$$\Sigma : \dot{x}(t) = A_k x(t) + A_d x(t - h(t)) \tag{6}$$

where Σ represents the overall system, and the matrix coefficients are:

$$A_k = \begin{bmatrix} A_1 - B_1 K_1 & 0 \\ 0 & A_2 - B_2 K_2 \end{bmatrix}; \quad A_d = \begin{bmatrix} 0 & A_{21} \\ A_{12} & 0 \end{bmatrix}; \quad x(t) = \begin{bmatrix} x_1(t) \\ x_2(t) \end{bmatrix} \tag{7}$$

The computation of the feedback gains for the cont roller is presented in the Appendix. The value of K_1 and K_2 are: is:

$$K_1(1) > \frac{\gamma_m - 1}{4} M_m \alpha^2 \tag{8}$$

$$K_1(2) > \gamma_m \alpha M_m + \frac{K_1(1)^2}{M_m} - B_m \tag{9}$$

$$K_2(1) > \frac{\gamma_s - 1}{4} M_s \alpha^2 \tag{10}$$

$$K_2(2) > \gamma_s \alpha M_s + \frac{K_1(1)^2}{M_s} - B_s \tag{11}$$

where $\alpha = \frac{1}{1-\dot{\tau}}$, $h(t) < \tau < 1$ **vi** represent s t he net work performa nce, and γ_m, γ_s are free design parameters. Since equations (9)-(1 1) have always a solution, it is possible to find a decentralized controller of t 11(1 form given by equations (4) that stabilizes a teleoperator with position-based force feedback in the presence of a variable communication delay.

This controller guarantees the stability of the master and the slave even in the case of interruption of the communication channel. In fact, $A_d = 0$ in equation (6) and the system is still stable since $A_k < 0$, from equation (1 8). This approach remains valid also when a position scale factor $K_s \neq 1$ is included in the forward path, since K_s must be compensated by a term $\frac{1}{K_s}$ in the feedback pat h, to preserve the consistency of t he force feedback. Finally, stability depends on the value of the time derivative of the delay, and not on the delay itself, thus providing an IOD stability condition [1 5].

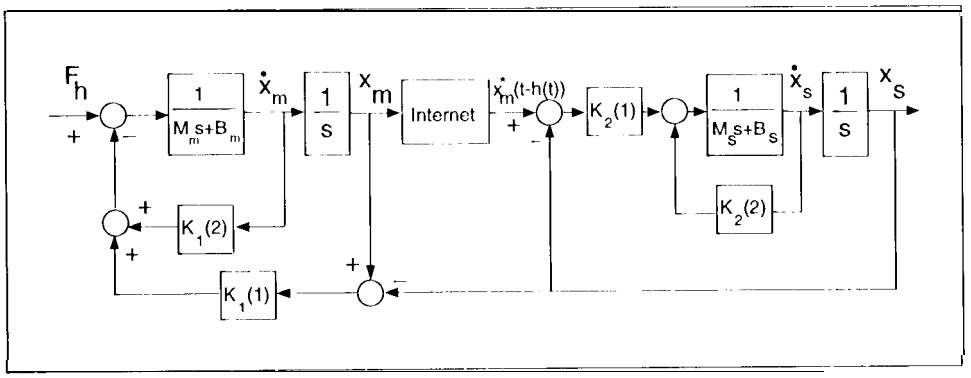

Figure 12: Block diagram of the state feedback controller.

4.2. **Jitter and losses compensation**

As mentioned in the previous Section, data sent over the Internet are corrupted by noise due to the time-(lclay jitter about its average value and packet losses, as shown earlier in Figure 7.

Delay jitter, however, is neither an independent additive noise, nor a phase noise, and it, affects both amplitude and frequency of the transmitted data. Since it is difficult to design a stochastic filter that eliminates this noise, we take advantage of the fact that haptic feedback is bandwidth and amplitude limited, and we approximate the true Internet noise with the *worst case* additive noise given by the following relation:

$$\sigma_a = (2\pi BA)^2 \sigma_d \qquad (12)$$

where A and B represent the maximum values of the amplitude and frequency of the transmitted signal, and σ_d is the variance of the delay computed from the RTT measurements.

We compensate this noise with an optimal filter at the slave side. The filter is an asymptotic Kalman filter, matching the master model and the input/output noise

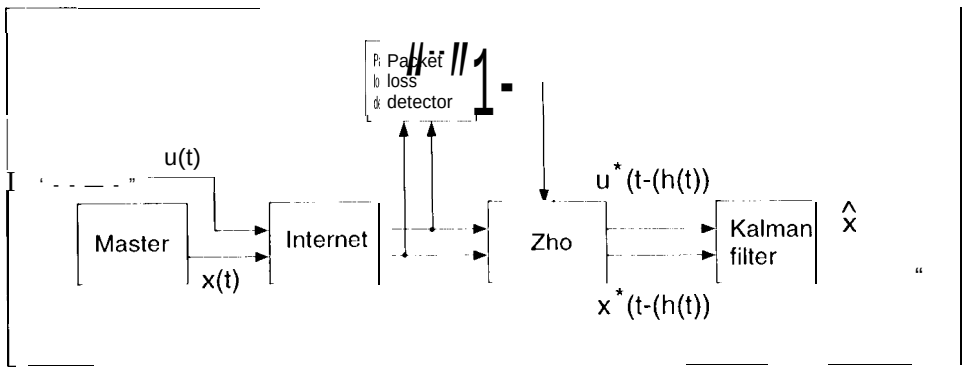

Figure 13: Proposed estimator-based jitter filtering.

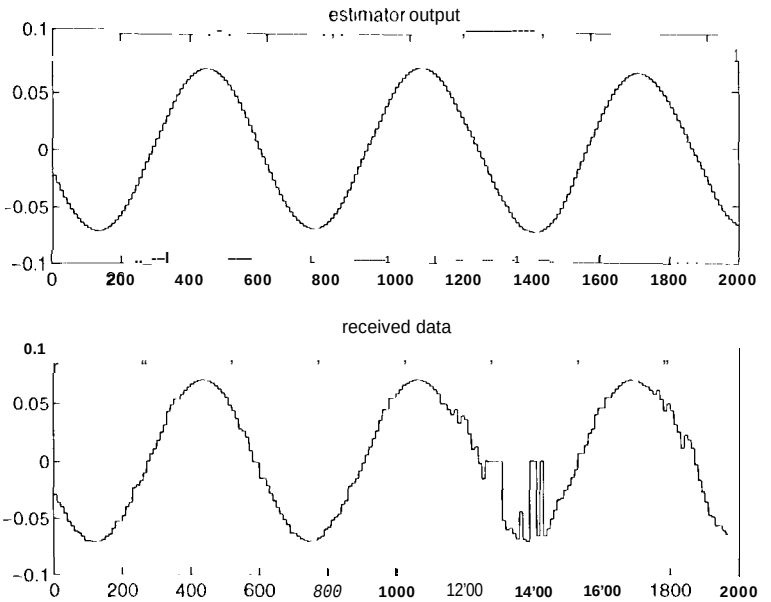

Figure 14: Comparison between received and estimated position.

variance of equation (1 2). The block diagram of the portion of the t elerobot with the jitter compensation estimator is shown in Figure 13.

Signals u^* and x^* in Figure 13 either t ake th e values of the delayed signals u and x wl ıen they are present, or keep the last values received, in Case of packet losses. In case of long interruptions then, the output of the estimator will converge to a position slightly different from the last received set point. This difference is specified during the filter design and must be within the safety margins of the system. The estimator filters short interruptions, such as single packet 1o s w s. The results of a simulation using this configuration is shown in Figure 14. The output of the filter is much smoother than the received signal, and thus acceptable as a reference for t he slave system.

The net force u applied by the operator t 0 the master represents the forces at the master handle, reduced by the force feedback generated by the master actuators. When a force sensor is not available, it is still possible to obtain a precise estimate by using an appropriate observer **[29].**

5. Examples

in this Section we present simulations and trots of the controller described in the previous Section with the experimental set-up of Figure 15. The Figure shows the real-tilllc portion of the design and control environment, represented by the laptop c omputer, and the 2-dof master. The laptop is a 486-DX66 PC, with two PCMCIA boards interfacing to the Internet and to the master. The master is the planar two-dof direct drive manipulator described in [9]. For the experiments, the controller cycle is set to $350\,Hz$.

'1'able 2 summarizes the characteristics of t he Internet segment between the master and the slave used in the simulations. Simulations cannot replicate the quality of the

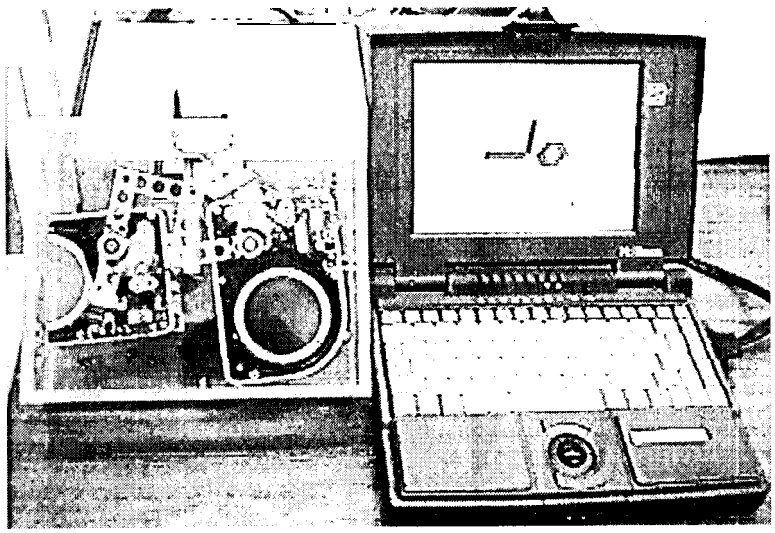

Figure 15: The Experimental set-up.

human perception, but they provide useful indications about the performance of the controller. The type of tuning possible is demonstrated with the plot of Figure 16, showing simulations carried out with mismatched controller gains. The top part of Figure 16 shows the position of the master and the slave, and the bottom part of the Figure shows the input force and the force feedback at the master. The response of the telerobot to a force input of $0.15N$, applied from $t = 0$ to $t = 2s$ is shown in the left side of the two plots. Due to the velocity feedback of both controllers, master and slave are rate limited. The force feedback generated at the master is due to the combined effect of position error and rate feedback. The force plot shows that the velocity feedback gain of the master is excessive, since the force feedback is saturated and stops with an overshoot as soon as the master stops moving, at $t = 2s$. At $t = 4s$ a force disturbance of $0.15N$ is applied to the slave. A small feedback force is generated at the master's side, due to the different values of the free parameter γ, set to 40 for the master and to 4 for the slave. In this case, the force feedback is only due to the position error, since the master is at rest. The simulation shows the need of adjusting the gains to better reproduce at the master the forces measured by the slave.

The final tuning of the controller is performed with experiments, such as those shown in Figures 17 and 18. These experiments consist of using a single link of the master as input, commanding the virtual manipulator to move against a hard surface. The

Host	Distance (km)	Avg Delay (ms)	Std. Dev.	Loss rate (%)	Sampling rate (ms)	Max $\dot{h}(t)$
Host 2	150	19	13.5	6.85	10	0.45

Table 2: Internet parameters used in the simulations

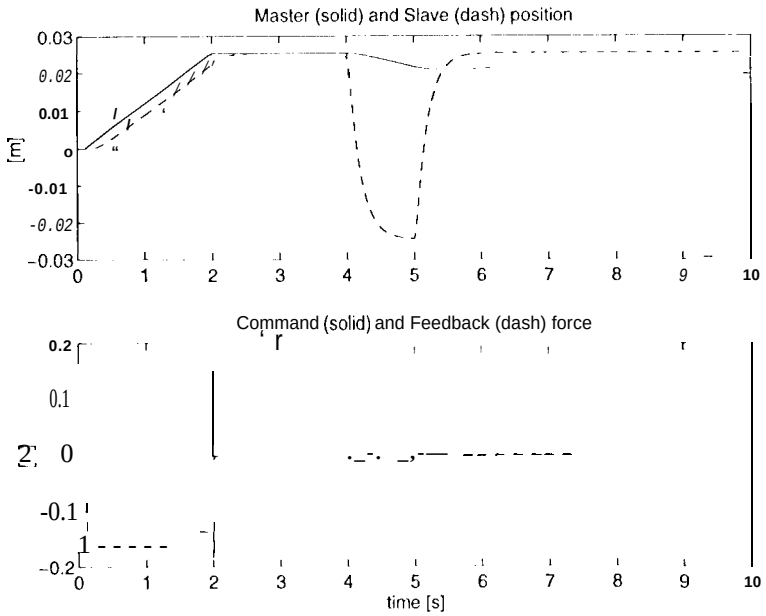

Figure 16: Simulation results.

mechanical parameters of the input device are as follows:

$$M_m = M_s = 0.05Kg; \quad B_m = B_s = 1Ns/m \tag{13}$$

The experiments are carried out over two Internet segments with the characteristics shown in Table 3. Since the two segments have different values Of the time derivative Of the delay, the resulting controllers are different, as shown in equations (9) and (9).

Figure 17 snows the response of our experimental telerobot when using a local reflector. The slave is commanded to push harder against the surface, by increasing the position of the master. The top part of Figure 17 shows the position s of both master and slave. The slave is blocked by the surface, and its position shows a small limit cycle due to the slow sampling rate. The bottom part of Figure 17 snows the force reflected at the master. The spikes are due to packet 1o sscs. A different type of response is shown in the plots of Figure 18, representing an experiment with a reflector located at a distance of 10,000 kin. In this example, a slightly larger position error generates a much lower force feedback. The longer time-delay results also in the slower slope of the force profile, and

Host	Distance (km)	Avg Delay (ms)	Std. Dev.	Loss rate (%)	Sampling rate (ms)	Max $\dot{h}(t)$
Host 1	.05	1.026	0.189	0.00	2.85	0.1
Host 3	10000	319.0	16.70	51.3	2.85	0.4

Table 3: Characteristics of the Internet segments used in the experiments.

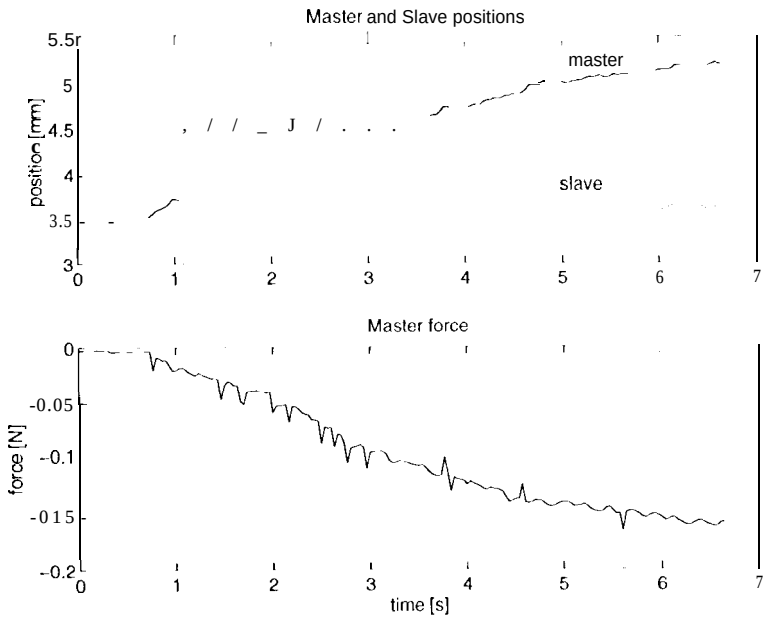

Figure 17: Experimental results wi 11 local connec ion.

in an apparent increase of the system compliance. The longer distance affects also the packet losses which are more frequent in this case.

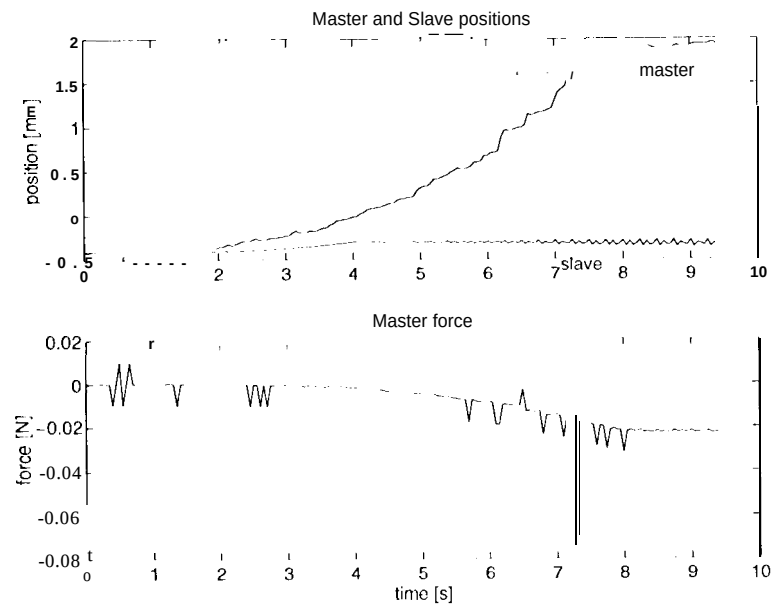

Figure 18: Experimental results with long distance connection.

6. Conclusion

A design environment for the identification, control design and test of a telerobotic system connected to an Internet segment is presented. It differs from previous design environments in that it is specifically designed to address the issues related to the control of' variable time-delay systems. The Internet segment connecting the two components of a telerobotic systems is treated as an unknown process introducing random delays and losses which are identified using probe packets. The delay parameters arc used in the design of the controller which is then tested in simulation and emulated using a two-dof force reflecting master and a virtual slave moving in a simulated world.

The underlying architecture includes *Matlab* identification and simulations modules, Internet communication routines, and a simple real-time kernel for emulating the telerobotic system. This implementation emphasizes low cost and portability to allow design and experiments with different systems and in different conditions. The real-time kernel does not require any specialized soft ware and it includes drivers for the hardware interface boards control the two-dof force reflecting master. The modular architecture provides a unified support for testing different controller structures and different delay compensation techniques.

The key features of this design environment include: (*i*) the off-line and on-line statistical identification of an Internet segment connecting the master and the slave of a telerobotic system; (*ii*) the design of a controller for a force-feedback telerobotic system compensating the Internet variable time-delay and packet losses; (*iii*) simulation of the controller with a *Matlab* package; and (*iv*) emulation of the system using a two-dof master connected to a simulated slave.

The functions of this environment are demonstrated in simulations and experiments providing qualitative information about the controller performance and the operator perception of the force-feedback. In the future we plan to move the environment to a more powerful computing platform and to enhance both the real-tilnc control and the graphical presentation. Parts of the current :\l'Cllit('('till'(! will be replaced by 11101e robust modules to ensure reliability of operation and more complex human-machine interaction.

7. Acknowledgment

The research described in this paper has been carried out at the Jet Propulsion Laboratory, California Institute of Technology, under a contract with the National Aeronautics and Space Administration, and at the University of Padova under grant # 203.07.25 from the National Research Council of Italy.

REFERENCES

1. R .J. Anderson. SM ART: A modular control architecture for telerobotics. *IEEE Roboticsand Automation Magazine*, 2(3) :1 018, September 1995.
2. R.J. Anderson and M.W. Spong. 1 3ilateral control of t eleoperators with time delay. *IEEE Transaction on A utomatic Control*, 34 (5) :494 501 , May 1989.

3. A. Banerjea, D. Ferrari, B. Mah, M. Moran, D. Verma, and H. Zhang. The tenet real-time protocol suite - design, implementation, and experiences. *IEEE ACM Transactions on Networking*, 4(1):1–10, February 1996.
4. J.C. Bolot. End-to-end packet delay and loss behavior in the internet. n *SIGCOMM '93*, pages 289–298, Ithaca, NY, September 1993.
5. J.C. Bolot, A.U. Shankar, and B.D. Plateau. Performance analysys of transport protocols over congestive networks. *Journal on Performance Evaluation*, (11):45–65, 1990.
6. R. Braden, D. Clark, and S. Shenker. Integrated services in the internet architecture: an overview. Technical Report RFC-1633, Network Working Group, June 1994.
7. P. Buttolo, J. Hewitt, R. Oboe, and B. Hannaford. Force feedback in virtual and shared environments. In *IEEE Conference on System, Man and Cybernetics*, Vancouver, BC, October 1995.
8. P. Buttolo, R. Oboe, and B. Hannaford. Architectures for shared haptic virtual environments. *Computer and Graphics*, 1997. accepted for publication.
9. P. Buttolo, R. Oboe, B. Hannaford, and McNeely B. Force feedback in shared virtual simulations. In *MICAD 96*, pages 157–162, Paris, France, February 1996.
10. H.H. Choi and M.J. Chung. Memoryless stabilization of uncertain dynamic systems with time-varying delayed states and controls. *Automatica*, 31(9):1349–1351, 1995.
11. D.E. Comer. *Interconnecting with TCP/IP*. Prentice Hall, 1991.
12. P.I. Corke. A robotics toolbox for MATLAB. *IEEE Robotics and Automation Magazine*, pages 24–32, March 1996.
13. H. Das et al. Operator performance with alternative manual control modes in teleoperation. *Presence*, 1(2):201–218, Spring 1992.
14. S. Deering and R. Hinden. Internet protocol, version 6 (ipv6) specification. Technical Report RFC-1833, Network Working Group, December 1995.
15. A. Eusebi and C. Melchiorri. Stability analysis of bilateral teleoperation robotic systems. In *3rd European Control Conference (ECC'95)*, Rome, Italy, 1995.
16. P. Fiorini and R. Oboe. Internet-based telerobotics: Problems and approaches. In *International Conference of Advanced Robotics (Icar'97)*, Monterey, CA, July 7-9 1997.
17. P. Fiorini, H. Seraji, and M. Long. A pc-based configuration controller for dexterous arms. *IEEE Robotics and Automation Magazine*, Fall 1997.
18. J.L. Hammond and P.J.P. O'Reilly. *Performance Analysis of Local Computer Networks*. Addison-Wesley, March 1988.
19. B. Hannaford and P. Fiorini. A detailed model of bilateral teleoperation. In *IEEE International Conference on Systems, Man and Cybernetics*, volume 2, pages 117–122, Beijin, China, August 1988.
20. K. Hirai and Y. Satoh. Stability of systems with variable time delay. *IEEE Transaction on Automatic Control*, ac-25(3):552–554, 1980.
21. W.S. Kim, B. Hannaford, and A.K. Bejczy. Force reflecion and shared compliant control in operating telemanipulators with time delay. *IEEE Transaction on Robotics and Automation*, 8(2):176–185, April 1992.
22. G. Leitmann. Guaranteed asymptotic stability for some linear system with bounded uncertainties. *Journal of Dynamic Systems, Measurements and Control*, 101:212,

1979.

23. R. Luck and A. Ray. Experimental verification of a delay compensation algorithm for integrated communication and control systems. *International Journal of Control*, 59(6):1357 1372, 1994.
24. R. Luck, A. nay, and Y. Halevi. Observability under recurrent loss of data. *AIAA Journal of Guidance, Control and Dynamics*, 15:284 287, 1992.
25. M.S. Mahmoud. Dynamic control of systems with variable st ate-delay. *International Journal of Robust and Nonlinear Control*, 6:123 146, 1 996.
26. S. McCanne and V. Jacobson. vie: a flexible framework for packet video. In *A CM Multimedia*, pages 1 12, San Francisco, CA, November 1995.
27. G . Niemeyer and J.E. Slotine. Stable adaptive teleoperation. *IEEE Journal of Oceanic Engineering*, 16(1):152 162, January 1991.
28. 1{. Oboe and P. Fiorini. Issues on internet -based teleoperat ion. In *Syroco 97*, Nantes, France, September 1997.
29. R. Oboe and K. Ohnishi. A space state approach to flexible robotic joint control. In *IECON 91 (Industrial Electronics Conference)*, Kobe, Japan, October 1991.
30. K.K Shyu and J .,1. Yan. Variable-structure model following adaptive control for systems with time-varying delay. *Control Theory and Advanced Technology, 10(3)* :513 521, September 1994.
31. D. Steward, D. Schmitz, and 1'. Khosla. Implementing real-tiln(: robotic systems using CHIMERA II. In *IEEE International Conference on R obotics and A utomation)*, pages 598 603, Sacramento, CA, March 1990.
32. H. Wu and K. Mizukami. Linear and nonlinear stabilizing continuous controllers of uncertain dynamical systems including state delay. *IEEE Transactions on Automatic Control, 41(1)*:116 121, January 1996.
33. H. Wu, K. Mizukami, and R. A. Wilgoss. Decentralized robust control for uncertain large-sca]c tilllc-delay dynamical systems. *Control Theory and Advanced Technology*, 9(2):533 546, June 1993.

Appendix: Controller Design

The design criteria for the decentralized feedback are found using the following Lyapunov functional:

$$V(x,t) = x(t)^T x(t) + \frac{1}{1-\bar{\tau}} \int_{t-h(t)}^{t} x(\theta)^T x(\theta) d\theta \geq 0 \tag{14}$$

with

$$\dot{h}(t) \leq \bar{\tau} < 1 \quad \forall t \tag{15}$$

The differentiation of equation (1 4) along t he solution of (6) yields:

$$\dot{V}(x, f) = 2x^T(A_k x + A_d x(t - \}/(f))) + \frac{1}{1-\bar{\tau}} x^T x + \frac{1 - h(t)}{1-\bar{\tau}} x(t - h(t))^T x(t - h(t)) \tag{16}$$

By using inequality

$$2x^T A_d x(t-h(t)) \leq x^T A_d A_d^T x + x(t-h(t))^T x(t-h(t)) \tag{17}$$

equation (16) becomes:

$$\dot{V}(x,t) \leq x^T(2A_k + A_dA_d^T + \frac{1}{1-\bar{\tau}}I_n)x + x(t - h(t))^T[I_n - \frac{1-h(t)}{1-\bar{\tau}}I_n]x(t - h(t)) \quad (18)$$

The asymptotic stability of equation (6) is guaranteed if $\dot{V}(x,t) < 0$. This inequality holds if the matrices in the two quadratic forms of the right side of equation (18) are both negative-definite. Given equation (15), the controller must satisfy:

$$S = 2A_k + A_dA_d^T + \frac{1}{1-\bar{\tau}}I_n < 0 \quad (19)$$

Condition (19) holds if all the eigenvalues of the matrix S are in $Re[s] < 0$, i.e.

$$Re\lambda_i(S) < 0 \quad \forall i \quad (20)$$

From equation (21), with $\alpha = \frac{1}{1-\bar{\tau}}$, it follows that matrix S has a block diagonal structure and, therefore, the eigenvalues of S are the eigenvalues of each block. Since both blocks have the same structure and each gain only appears in the block relative to its controller, the solution of equation (20) is significantly simplified, and it is possible to design independent controllers for master and slave.

The matrix S is as follows:

$$S = \begin{bmatrix} \alpha & 2 & 0 & 0 \\ -2(\frac{K_1(1)}{M_m}) & -2(\frac{K_1(2)+B_m}{M_m}) + (\frac{K_1(1)}{M_m})^2 + \alpha & 0 & 0 \\ 0 & 0 & \alpha & 2 \\ 0 & 0 & -2(\frac{K_2(1)}{M_s}) & -2(\frac{K_2(2)+B_s}{M_s}) + (\frac{K_2(1)}{M_s})^2 + \alpha \end{bmatrix} \quad (21)$$

The upper-left block of S can be rewritten as follows:

$$S_1 = \begin{bmatrix} \alpha & 2 \\ -\xi_{11} & -\xi_{21} + \alpha \end{bmatrix}; \quad \xi_{11} = \frac{2K_1(1)}{M_m}, \quad \xi_{21} = 2(\frac{K_1(2)+B_m}{M_m}) - (\frac{K_1(1)}{M_m})^2 \quad (22)$$

The eigenvalues of S_1 are computed using its characteristic polynomial:

$$det(sI_2 - S_1) = s^2 + s(\xi_{21} - 2\alpha) + 2\xi_{11} + \alpha^2 - \xi_{21}\alpha \quad (23)$$

Equation (23) has roots with negative real part if the two rightmost terms are both positive:

$$Re(\lambda_i(S_1)) < 0 \leftrightarrow \begin{cases} 2\xi_{11} + \alpha^2 - \xi_{21}\alpha > 0 \\ \xi_{21} - 2\alpha > 0 \end{cases} \quad (24)$$

The following stability conditions are then derived:

$$\begin{cases} \xi_{21} = \gamma_m\alpha \\ \xi_{11} > \frac{\gamma_m+1}{2}\alpha^2 \end{cases}; \quad \gamma_m > 2 \quad (25)$$

where γ_m is a free design parameter in the master controller, that influences the performance of the teleoperator.

Similarly, for the slave controller, the stability conditions are:

$$\begin{cases} \xi_{22} = \gamma_s\alpha \\ \xi_{12} > \frac{\gamma_s+1}{2}\alpha^2 \end{cases}; \quad \gamma_s > 2 \quad (26)$$

The gains given by equations (9) to (11) are then computed by manipulating equations (25) and (26).